

A random version of principal component analysis in data clustering

Luigi Leonardo Palese^{1,*},

1 University of Bari "Aldo Moro", Department of Basic Medical Sciences, Neurosciences and Sense Organs (SMBNOS), Bari, 70124, Italy

* luigileonardo.palese@uniba.it

Abstract

Principal component analysis (PCA) is a widespread technique for data analysis that relies on the covariance-correlation matrix of the analyzed data. However to properly work with high-dimensional data, PCA poses severe mathematical constraints on the minimum number of different replicates or samples that must be included in the analysis. Here we show that a modified algorithm works not only on well dimensioned datasets, but also on degenerated ones.

Introduction

Science today is surrounded by large amounts of data. These are produced by techniques and instruments able to measure a huge number of variables on a large number of samples, they are deposited in an increasing number of online databases that grow exponentially, and also modern numerical simulations can produce very large and high-dimensional outputs. The challenge of the growing size of data concerns all sciences, but the field in which we have seen the most spectacular growth is probably that of life sciences, where the advancement of genomics, proteomics and high-throughput technologies has produced an overwhelming amount of data, more and more often freely available to all researchers. Beside the large number of samples, these data are big also because they are high-dimensional: this means that each sample, or instance, of a typical dataset contains a large number of degree of freedom. Such high-dimensionality makes visualization and exploration of samples and datasets very difficult. To overcome these limitations, a series of techniques have been developed that help researchers in visualization, exploration and mining of large data. [44]

Among the various algorithms that reduce the dimensionality of data, while retaining the important information, one of the most successful is principal component analysis (PCA). [36] PCA has been reinvented several times, but it has been developed in its modern form by Pearson and Hotelling. [8,19,31] How PCA works is recalled in the Methods section, but here it is important to note that, in its classical implementation, PCA relies on the covariance (or also correlation) matrix of the analyzed data. This is actually a point often overlooked by end-users, but it should be stressed that the number of samples needed to accurately estimate the covariance/correlation matrix of a system containing n degree of freedom should be (much) larger than n . Otherwise the covariance/correlation matrix will be full of spurious correlations, as well as rank deficient from a mathematical point of view if the

number of samples is less than n . However here we will show that what is important for the functioning of the method it is not a particular covariance/correlation matrix, but rather the symmetry that characterizes this type of matrices.

Results and Discussion

We developed the method in order to calculate the PCA of a set of protein crystallographic structures. Proteins are structurally and dynamically complex objects. [13,27] Because molecular dynamics (MD) is actually at a level of accuracy [10] that permits to predict experimentally observables, [6] it is nowadays a standard tool for the dynamical characterization of proteins. In the analysis of MD trajectories PCA is of widespread use: [23] the high-dimensional large number of different molecular conformations that constitute the output of a MD experiment is an ideal dataset for PCA. [7] On the other hand, the number of protein structures reported in the Protein Data Bank [2] (PDB) is collectively large, but there are few structures of a single protein. Although it is possible to find dozens or even hundreds of versions of a single protein in the PDB, the number of available structures is incomparably smaller than the number of degree of freedom of a protein. So while PCA can be used in the analysis of the thousands of conformations obtained from MD simulations, in its classical implementation PCA can not be used in the analysis of the experimental structures as the low number of different conformations reported in the PDB does not allow an accurate calculation of the covariance matrix. To perform a structural analysis similar to the PCA we choose to analyze the human serum albumin (HSA) available structures in the PDB. HSA, [12] the most abundant protein in plasma, is a monomeric multi-domain molecule. HSA is a non-glycosylated, all- α protein chain of 65 kDa, with a globular heart-shaped conformation consisting of three homologous domains (I-III). Each domain is composed by two subdomains (A and B). It is an important transport protein with different binding sites able to accommodate a number of chemically different ligands. HSA represents the main carrier for fatty acids (there are seven binding sites for fatty acids, labeled as FA1 to FA7), and it is a depot and carrier for exogenous compounds (mainly, but not exclusively at the Sudlow's sites I and II), thus affecting the pharmacokinetics of many drugs.

Among the available structures, we selected 58 structure for the analysis (see Methods for selection criteria). After structural alignment, the α -carbon atom Cartesian coordinates were extracted and arranged in a data matrix in which each row represented a single HSA structure. Thus the data matrix was composed of 58 rows and 1695 columns (565 α -carbon atoms were finally included in the analysis; see the Methods section). This is clearly a degenerated dataset, as it is impossible to obtain the true correlation matrix of a multivariate system with 1695 degree of freedom by using only 58 samples. If we calculate the correlation matrix, this will be, at best, only a rank deficient approximation of the true one in which a large number of false correlations must be expected. While it is true that, using a careful error handling (and silencing) program, or also using algorithms that estimate the principal components without ever computing the covariance matrix, [38] it is generally possible to calculate the first principal components, the classical PCA is not calculable on this dataset.

It should be considered that what really we are interested in is not the identification of the axes that describe the greatest variance of the data (axes which do not have a particular *a priori* meaning), but instead an orthogonal linear transformation of data that could be useful in exploratory data analysis. We can relax the request that the correlation-covariance matrix (the true or the approximated one) is needed for such transformation: it is possible that what is important in PCA as clustering tool may not be the use of a *particular* matrix, but instead of a matrix belonging to a particular

symmetry class. The bases for such an hypothesis are rooted in the fact that good models for the covariance matrices for the protein configurations obtained from MD [28–30] are a class of symmetric random matrices. [11] Moreover, the consequences of the Johnson-Lindenstrauss lemma, [22] and the fact that in the Pearson original view [8, 31] of PCA which is important is the subspace and not the axes as such, furnish us a further justification. Thus we applied to the albumin dataset a variant of the PCA algorithm, in which a square symmetric random matrix was used, instead of the correlation-covariance one (this matrix was obviously of dimension of 1695). We will refer to this algorithm as random component analysis (RCA). The detailed algorithm is described in the Methods section, and an easily customizable implementation is reported in Supplementary Information section.

The results of this analysis are reported in Figure 1. As can be easily appreciated by inspecting the figure, RCA leads to two well defined clusters of structures, and what is more interesting is that one cluster contains all and only the HSA molecules with bound fatty acid, the other one only structures without fatty acid. These cluster are

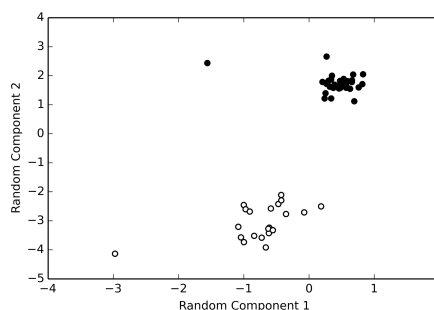


Figure 1. Random component analysis of the HSA structures. The Figure reports a random component analysis on the HSA structures contained in the dataset described in the text. The HSA structures with bound fatty acids are reported as solid (black) circles, whereas the structures without bound fatty acids are reported as void (white) circles. The algorithm clearly permits to differentiate two clusters of structures in the dataset, and the discriminant is the presence of absence, respectively, of bound fatty acids. Two similar cluster of structures have been obtained in all the random component analysis calculations carried out on the HSA dataset (see Supplementary Information).

reproducible (see Supplementary Figure 1) and are similar to those obtained by different protocols (see Methods and Supplementary Figure 2). It worth noting that a large number of structural and functional works on HSA lead to the conclusion that two structures, possibly related to the presence of fatty acids, are discernible for this protein. [1, 12] Our RCA analysis permits to go further, as it clearly demonstrates that the only discriminant for such structural switch in the whole dataset is the presence or absence of bound fatty acid.

While RCA has been developed for degenerated datasets (that means datasets that are characterized by a larger number of degree of freedom respect to the number of samples that can be analyzed) we tested it also on well sized datasets. These were retrieved the from the UCI (University of California at Irvine, School of Information and Computer Science) Machine Learning Repository (Lichman, M., <http://archive.ics.uci.edu/ml>). The results of PCA and RCA on the classical Iris dataset are reported in Figure 2. It can be stated that RCA is at least not inferior to PCA in clustering purposes, and the results are reproducible, as reported in Supplementary Figure 3. Similar results have been obtained with the Wine dataset, which are reported in the Supplementary Figure 4.

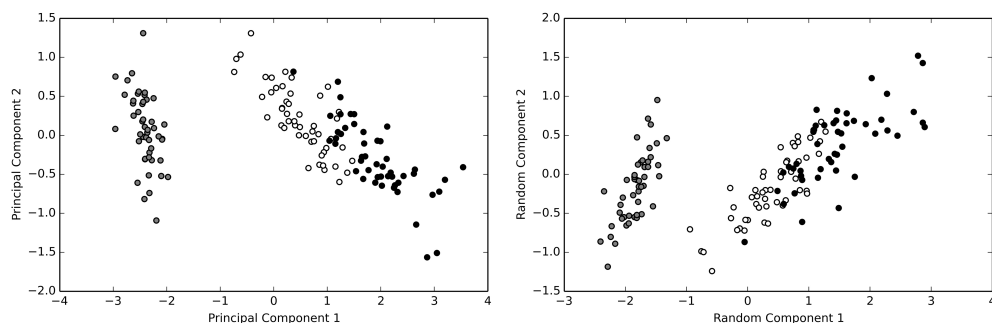


Figure 2. The Iris dataset. Principal component analysis (left) and random component analysis (right) of the Iris dataset are reported. The Iris dataset is a simple but classical benchmark for the clustering algorithms. This dataset contains 150 entries, 50 for each of the species *Iris virginica* (black), *Iris setosa* (gray) and *Iris versicolor* (white). Both algorithms easily differentiate the *Iris setosa* cluster, whereas the other species can be only partially discriminated by the algorithms of this class. The full set of random component analysis carried out on the Iris dataset shows similar clustering results and it is reported in Supplementary Information.

Conclusions

The algorithm proposed in this communication is easy to implement, conceptually simple and numerically robust. It is another example of useful application of random matrix theory, [11,28–30] whose pervasiveness is even more evident in a large number of fields. This work demonstrates that what is important for clustering efficiency of PCA is not the exact form of the covariance-correlation matrix, but instead simply its symmetry, as in our RCA algorithm. The fact that good and informative clustering can be achieved by random projection is nowadays an emerging concept that, beside practical applications, could have far reaching implications also from a conceptual point of view. Finally, this work suggests that an excessive confidence on correlations (which are often spurious) and on large covariance should be avoided, if a simple random matrix could well surrogate them in cluster generation.

Methods

HSA database construction

In order to build up a suitably large dataset of structures we searched in the Protein Data Bank [2] (www.rcsb.org) for the albumin structures, with the constraints of specie (human), single protein type in the structure, and resolution of 3.30 Å or better. After this initial screening, because some N- and C-terminal residues are often not present in the deposited structure, and in order to include the largest possible number of structures as complete as possible, the ones starting after the SER 5 and ending before ALA 569 were excluded from the database. Finally, the structures containing a number of α -carbon atoms different of 565 were also excluded. The final dataset contained 58 structures [3–5, 9, 14–18, 33–35, 40, 42, 43, 46–52] which are reported in the Supplementary Table 1.

A pdb file for each of these structures has been written in VMD [20] (from SER 5 to ALA 569); these structures were aligned using MultiSeq [37] and the pdb files were updated to the new coordinates. The same software was used to calculate the distance trees (RMSD and Qh style). [25,39] The obtained tree are reported as Supplementary

Figure 2 (see also the Supplementary Table 2).

To obtain the dataset in a matrix form, the pdb files were loaded in VMD and the α -carbon atom coordinates were extracted and written in a text file such that each row described a structure, by a Tcl (www.tcl.tk) script. Curly brackets in the raw text file were eliminated by vim scripting (www.vim.org), so as to obtain the data matrix in a readable file format by the numerical analysis software.

The PCA and RCA algorithms

PCA was based on the eigenvector decomposition of the correlation matrix. [7, 8, 28, 29, 36, 41, 44] After the centroid subtraction, the covariance matrix of the dataset matrix described above was obtained as

$$C_{ij} = \langle (x_i - \langle x_i \rangle) (x_j - \langle x_j \rangle) \rangle$$

where $\langle \dots \rangle$ represents the average over all the conformations in the dataset. The correlation matrix was calculated from this matrix as

$$P_{ij} = \frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}}$$

with obvious meaning of symbols. This square symmetric matrix was diagonalized

$$R^T P R = \Lambda$$

using standard numerical routines (see below), where R is an orthonormal transformation matrix, the superscript T means transposition and Λ is a diagonal matrix whose elements are the eigenvalues. The empirical matrix was projected onto the eigenvectors to give the principal components.

The RCA was performed exactly as the PCA, except for the fact that the square symmetric correlation matrix was replaced by a random symmetric matrix, obtained as

$$M = \frac{G + G^T}{2}$$

where G was a normal distributed random square matrix. So this algorithm could be conceived as a version of classical PCA with relaxed constraints respect to the matrix to be used in calculating the new orthonormal reference system, where only the matrix symmetry is preserved.

Software implementation and code availability

The PCA and RCA algorithms were implemented in the Python language (www.python.org) in an IPython notebook. [32] The NumPy numerical software library [45] was used, which is part of the Scipy [26] software package. The Pandas [24] and Matplotlib [21] packages were used to import the Iris and Wine datasets and to obtain the all graphical outputs, respectively (both packages were obtained from Scipy; www.scipy.org). The implementation of these algorithms is reported in Supplementary Information in Python format. Two versions of the RCA algorithm are reported: the first one requires the dataset and the dimension of the dummy correlation matrix as arguments, while the second requires as arguments the dataset and the random matrix that will be used for the calculation of the orthogonal projection system. This last function could be useful if one would save a particularly interesting matrix for further analysis. These files are easily customizable; as it is provided, the software requires (very) few seconds for the download and analysis of the proposed datasets (the HSA

dataset described above, the Iris and Wine datasets) on an Intel Core i7 machine or a Xeon equipped workstation, both running Ubuntu 14.04 LTS. Very large datasets (as in the case of MD outputs; not shown) could require up to (also several) minutes to be analyzed. Because the RCA algorithm performs a random projection it is preferable to carry out multiple runs of it. In a small percentage of cases (no more than 5% - 10% of the tests used for this work) the algorithm does not get a projection that separates the samples in different clusters, and this is the only drawback of the simple implementation of the RCA algorithm here described.

References

1. P. Ascenzi and M. Fasano. Allostery in a monomeric protein: the case of human serum albumin. *Biophys. Chem.*, 148(1):16–22, 2010.
2. H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Res.*, 28(1):235–242, 2000.
3. A. A. Bhattacharya, S. Curry, and N. P. Franks. Binding of the general anesthetics propofol and halothane to human serum albumin high resolution crystal structures. *J. Biol. Chem.*, 275(49):38731–38738, 2000.
4. A. A. Bhattacharya, T. Grüne, and S. Curry. Crystallographic analysis reveals common modes of binding of medium and long-chain fatty acids to human serum albumin. *J. Mol. Biol.*, 303(5):721–732, 2000.
5. A. Bijelic, S. Theiner, B. K. Keppler, and A. Rompel. X-ray structure analysis of indazolium trans-[tetrachlorobis (1h-indazole) ruthenate (iii)](kp1019) bound to human serum albumin reveals two ruthenium binding sites and provides insights into the drug binding mechanism. *J. Med. Chem.*, 2016.
6. F. Bossis and L. L. Palese. Molecular dynamics in cytochrome c oxidase mössbauer spectra deconvolution. *Biochem. Biophys. Res. Commun.*, 404(1):438–442, 2011.
7. F. Bossis and L. L. Palese. Amyloid beta (1–42) in aqueous environments: effects of ionic strength and e22q (dutch) mutation. *Biochim. Biophys. Acta*, 1834(12):2486–2493, 2013.
8. R. Bro and A. K. Smilde. Principal component analysis. *Anal. Methods*, 6(9):2812–2831, 2014.
9. D. Buttar, N. Colclough, S. Gerhardt, P. A. MacFaul, S. D. Phillips, A. Plowright, P. Whittamore, K. Tam, K. Maskos, S. Steinbacher, et al. A combined spectroscopic and crystallographic approach to probing drug–human serum albumin interactions. *Bioorg. Med. Chem.*, 18(21):7486–7496, 2010.
10. R. O. Dror, R. M. Dirks, J. Grossman, H. Xu, and D. E. Shaw. Biomolecular simulation: a computational microscope for molecular biology. *Annu. Rev. Biophys.*, 41:429–452, 2012.
11. A. Edelman and Y. Wang. Random matrix theory and its innovative applications. In *Advances in Applied Mathematics, Modeling, and Computational Science*, pages 91–116. Springer, 2013.

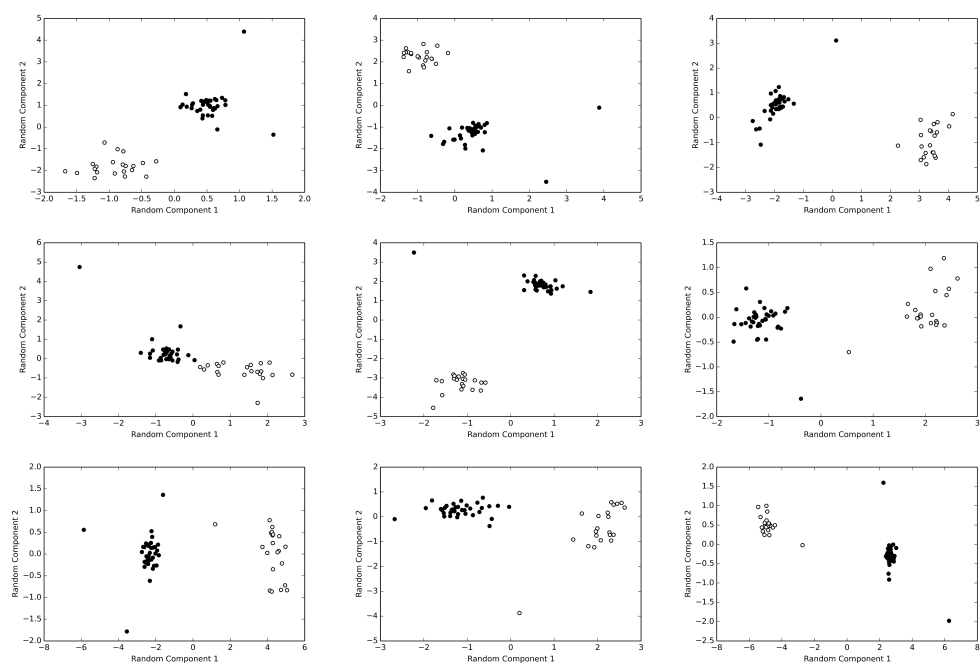
-
12. G. Fanali, A. di Masi, V. Trezza, M. Marino, M. Fasano, and P. Ascenzi. Human serum albumin: from bench to bedside. *Mol. Aspects Med.*, 33(3):209–290, 2012.
 13. H. Frauenfelder. Proteins: paradigms of complexity. *Proc. Natl. Acad. Sci. U.S.A.*, 99(suppl 1):2479–2480, 2002.
 14. J. Ghuman, P. A. Zunszain, I. Petitpas, A. A. Bhattacharya, M. Otagiri, and S. Curry. Structural basis of the drug-binding specificity of human serum albumin. *J. Mol. Biol.*, 353(1):38–52, 2005.
 15. S. Guo, X. Shi, F. Yang, L. Chen, E. J. Meehan, C. Bian, and M. Huang. Structural basis of transport of lysophospholipids by human serum albumin. *Biochem. J.*, 423(1):23–30, 2009.
 16. X. M. He and D. C. Carter. Atomic structure and chemistry of human serum albumin. *Nature*, 358(6383):209–215, 1992.
 17. Y. He, T. Ning, T. Xie, Q. Qiu, L. Zhang, Y. Sun, D. Jiang, K. Fu, F. Yin, W. Zhang, et al. Large-scale production of functional human serum albumin from transgenic rice seeds. *Proc. Natl. Acad. Sci. U. S. A.*, 108(47):19078–19083, 2011.
 18. K. L. Hein, U. Kragh-Hansen, J. P. Morth, M. D. Jeppesen, D. Otzen, J. V. Møller, and P. Nissen. Crystallographic analysis reveals a unique lidocaine binding site on human serum albumin. *J. Struct. Biol.*, 171(3):353–360, 2010.
 19. H. Hotelling. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.*, 24(6):417, 1933.
 20. W. Humphrey, A. Dalke, and K. Schulten. Vmd: visual molecular dynamics. *J. Mol. Graphics*, 14(1):33–38, 1996.
 21. J. D. Hunter et al. Matplotlib: A 2d graphics environment. *Comput. Sci. Eng.*, 9(3):90–95, 2007.
 22. W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Cont. Math.*, 26(189-206):1, 1984.
 23. A. Kitao and N. Go. Investigating protein dynamics in collective coordinate space. *Curr. Opin. Struct. Biol.*, 9(2):164–169, 1999.
 24. W. McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56, 2010.
 25. P. O’Donoghue and Z. Luthey-Schulten. Evolutionary profiles derived from the qr factorization of multiple structural alignments gives an economy of information. *J. Mol. Biol.*, 346(3):875–894, 2005.
 26. T. E. Oliphant. Python for scientific computing. *Comput. Sci. Eng.*, 9(3):10–20, 2007.
 27. L. L. Palese. Protein dynamics: complex by itself. *Complexity*, 18(3):48–56, 2013.
 28. L. L. Palese. Correlation analysis of trp-cage dynamics in folded and unfolded states. *J. Phys. Chem. B*, 119(51):15568–15573, 2015.
 29. L. L. Palese. Random matrix theory in molecular dynamics analysis. *Biophys. Chem.*, 196:1–9, 2015.

-
30. L. L. Palese. Protein states as symmetry transitions in the correlation matrices. *J. Phys. Chem. B*, (in press), 2016.
 31. K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *Philos. Mag.*, 2(11):559–572, 1901.
 32. F. Pérez and B. E. Granger. Ipython: a system for interactive scientific computing. *Computing in Science & Engineering*, 9(3):21–29, 2007.
 33. I. Petitpas, A. A. Bhattacharya, S. Twine, M. East, and S. Curry. Crystal structure analysis of warfarin binding to human serum albumin anatomy of drug site i. *J. Biol. Chem.*, 276(25):22804–22809, 2001.
 34. I. Petitpas, T. Grüne, A. A. Bhattacharya, and S. Curry. Crystal structures of human serum albumin complexed with monounsaturated and polyunsaturated fatty acids. *J. Mol. Biol.*, 314(5):955–960, 2001.
 35. I. Petitpas, C. E. Petersen, C.-E. Ha, A. A. Bhattacharya, P. A. Zunszain, J. Ghuman, N. V. Bhagavan, and S. Curry. Structural basis of albumin–thyroxine interactions and familial dysalbuminemic hyperthyroxinemia. *Proc. Natl. Acad. Sci. U. S. A.*, 100(11):6440–6445, 2003.
 36. M. Ringnér. What is principal component analysis? *Nat. biotechnol.*, 26(3):303–304, 2008.
 37. E. Roberts, J. Eargle, D. Wright, and Z. Luthey-Schulten. Multiseq: unifying sequence and structure data for evolutionary analysis. *BMC Bioinformatics*, 7(1):1, 2006.
 38. S. Roweis. Em algorithms for pca and spca. *Adv. Neural Inf. Process. Syst.*, pages 626–632, 1998.
 39. R. B. Russell and G. J. Barton. Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins*, 14(2):309–323, 1992.
 40. A. J. Ryan, J. Ghuman, P. A. Zunszain, C.-w. Chung, and S. Curry. Structural basis of binding of fluorescent, site-specific dansylated amino acids to human serum albumin. *J. Struct. Biol.*, 174(1):84 – 91, 2011.
 41. J. Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.
 42. A. Sivertsen, J. Isaksson, H.-K. S. Leiros, J. Svenson, J.-S. Svendsen, and B. O. Brandsdal. Synthetic cationic antimicrobial peptides bind with their hydrophobic parts to drug site ii of human serum albumin. *BMC Struct. Biol.*, 14(1):1, 2014.
 43. S. Sugio, A. Kashima, S. Mochizuki, M. Noda, and K. Kobayashi. Crystal structure of human serum albumin at 2.5 Å resolution. *Protein Eng.*, 12(6):439–446, 1999.
 44. L. Van Der Maaten, E. Postma, and J. Van den Herik. Dimensionality reduction: a comparative. *J. Mach. Learn. Res.*, 10:66–71, 2009.
 45. S. Van Der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: a structure for efficient numerical computation. *Comput. Sci. Eng.*, 13(2):22–30, 2011.

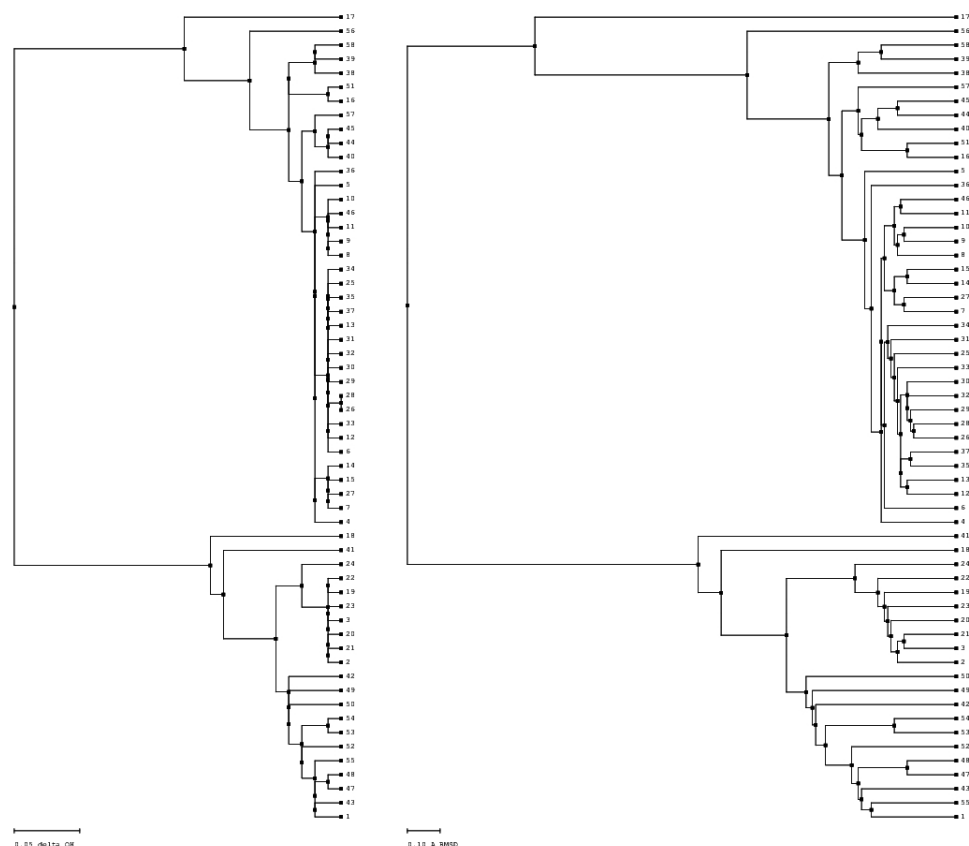
-
46. Y. Wang, H. Yu, X. Shi, Z. Luo, D. Lin, and M. Huang. Structural mechanism of ring-opening reaction of glucose by human serum albumin. *J. Biol. Chem.*, 288(22):15980–15987, 2013.
 47. Z.-m. Wang, J. X. Ho, J. R. Ruble, J. Rose, F. Rüker, M. Ellenburg, R. Murphy, J. Click, E. Soistman, L. Wilkerson, et al. Structural studies of several clinically important oncology drugs in complex with human serum albumin. *Biochim. Biophys. Acta*, 1830(12):5356–5374, 2013.
 48. M. Wardell, Z. Wang, J. X. Ho, J. Robert, F. Ruker, J. Ruble, and D. C. Carter. The atomic structure of human methemalbumin at 1.9 Å. *Biochem. Biophys. Res. Commun.*, 291(4):813–819, 2002.
 49. F. Yang, C. Bian, L. Zhu, G. Zhao, Z. Huang, and M. Huang. Effect of human serum albumin on drug metabolism: structural evidence of esterase activity of human serum albumin. *J. Struct. Biol.*, 157(2):348–355, 2007.
 50. Y. Zhang, P. Lee, S. Liang, Z. Zhou, X. Wu, F. Yang, and H. Liang. Structural basis of non-steroidal anti-inflammatory drug diclofenac binding to human serum albumin. *Chem. Biol. Drug Des.*, 86(5):1178–1184, 2015.
 51. L. Zhu, F. Yang, L. Chen, E. J. Meehan, and M. Huang. A new drug binding subsite on human serum albumin and drug–drug interaction studied by x-ray crystallography. *J. Struct. Biol.*, 162(1):40–49, 2008.
 52. P. A. Zunszain, J. Ghuman, T. Komatsu, E. Tsuchida, and S. Curry. Crystal structural analysis of human serum albumin complexed with heme and fatty acid. *BMC Struct. Biol.*, 3(1):6, 2003.

Supplementary Information

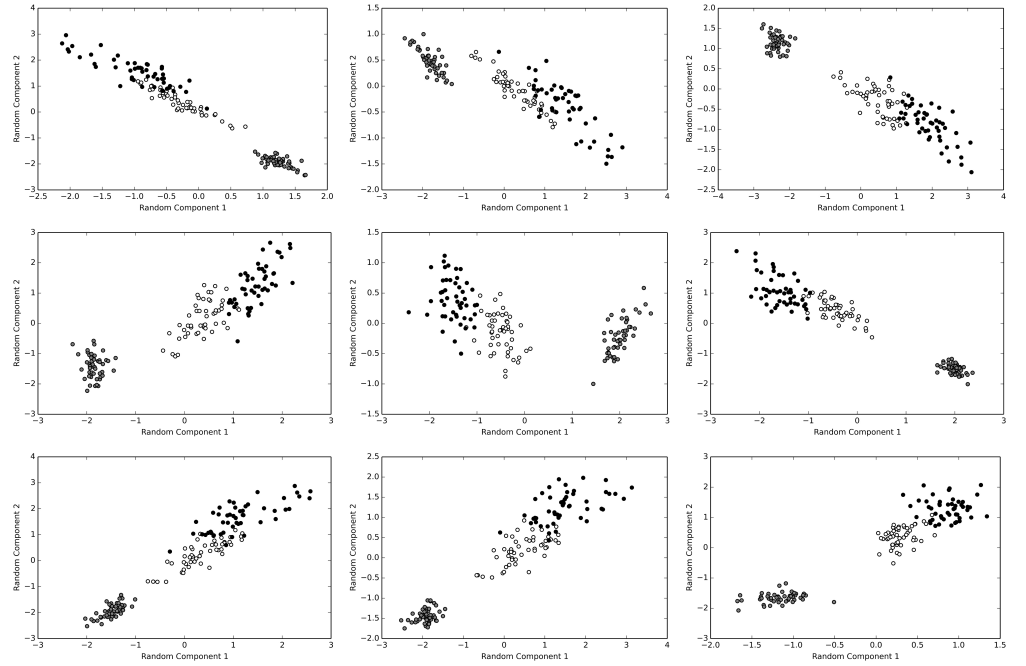
Supplementary figures and code



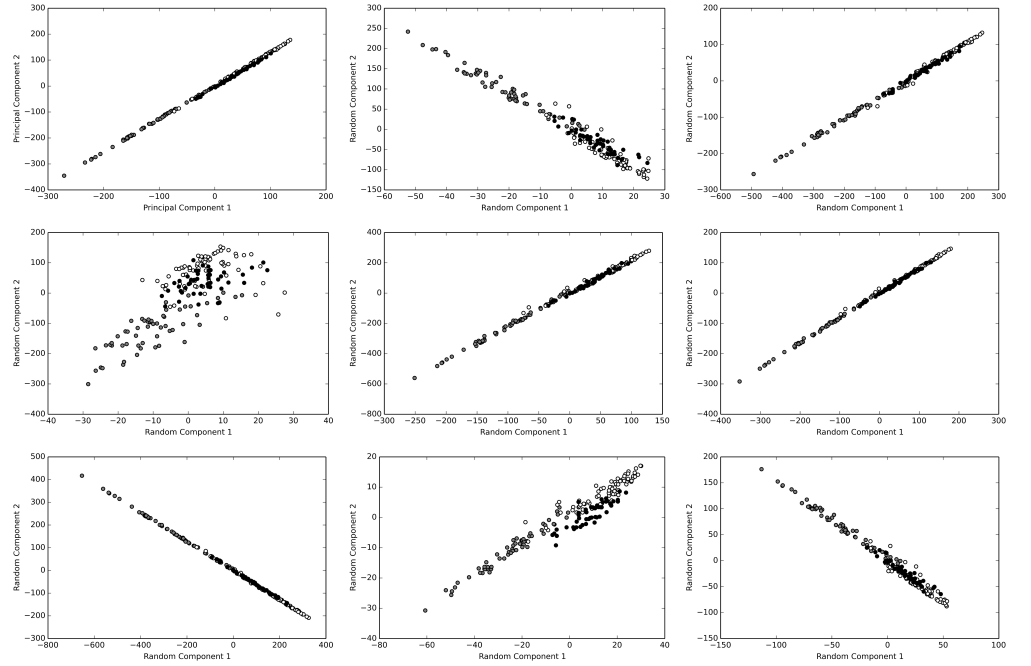
Supplementary Figure 1. HSA random component analysis. The figure reports 9 consecutive application runs of the random component analysis algorithm on the HSA dataset described in the main text. The HSA structures with bound fatty acids are reported as black circles, whereas the HSA structures without fatty acids (either ligand free or with chemically different ligands) are reported as open (white) circles. It is evident that the two class of HSA structures are in different clusters.



Supplementary Figure 2. Tree clustering of HSA dataset. The Qh (left) and RMSD (right) trees have been obtained with the MultiSeq program as described in the main text. The numbers correspond to the protein PDB entries as reported in the Supplementary Table 1 (see below). Both algorithms recognize two clusters (cluster A and B in Supplementary Table 2): cluster A contains the HSA structures without bound fatty acids, while cluster B contains HSA molecules with bound fatty acid.



Supplementary Figure 3. Random component analysis of Iris dataset. The figure reports 9 consecutive application runs of the random component analysis algorithm on the Iris dataset. The *Iris virginica* species is reported in black, the *Iris setosa* in gray and *Iris versicolor* in white. As in the case of PCA, the algorithm easily differentiate the *Iris setosa* cluster, whereas the other species can be only partially discriminated by the algorithms of this class (linear classifiers).



Supplementary Figure 4. The wine dataset. The dataset was retrieved from the UCI repository (see text). It contains three different cultivars, indicated as 1, 2 and 3 in the on line repository and here reported as gray, white and black circles. The top-left panel reports the principal component analysis of this dataset, while all other panels report different runs of random component analysis. The cultivars, particularly those indicated as 2 and 3, partially overlap with both algorithms.

#	PDB	Ligands	Res A
1	1BM0	N/A	2,5
2	1E78	N/A	2,6
3	1E7A	2,6-BIS(1-METHYLETHYL)PHENOL (PROPOFOL)	2,2
4	1E7C	MYRISTIC ACID; 2-BROMO-2-CHLORO-1,1,1-TRIFLUOROETHANE	2,4
5	1E7E	DECANOIC ACID	2,5
6	1E7F	LAURIC ACID	2,43
7	1E7G	MYRISTIC ACID	2,5
8	1E7H	PALMITIC ACID	2,43
9	1E7I	STEARIC ACID	2,7
10	1GNI	OLEIC ACID	2,4
11	1GNJ	ARACHIDONIC ACID	2,6
12	1H9Z	R-WARFARIN; MYRISTIC ACID	2,5
13	1HA2	S-WARFARIN; MYRISTIC ACID	2,5
14	1HK4	3,5,3',5'-TETRAIODO-L-THYRONINE; MYRISTIC ACID	2,4
15	1HK5	3,5,3',5'-TETRAIODO-L-THYRONINE; MYRISTIC ACID	2,7
16	1N5U	PROTOPORPHYRIN IX CONTAINING FE (HEME); MYRISTIC ACID	1,9
17	1O9X	PROTOPORPHYRIN IX CONTAINING FE (HEME); MYRISTIC ACID	3,2
18	1UOR	N/A	2,8
19	2BX8	AZAPROPAZONE	2,7
20	2BXB	4-BUTYL-1-(4-HYDROXYPHENYL)-2-PHENYLPYRAZOLIDINE- 3,5-DIONE (OXYPHENBUTAZONE)	3,2
21	2BXC	4-BUTYL-1,2-DIPHENYL-PYRAZOLIDINE-3,5-DIONE	3,1
22	2BXD	R-WARFARIN	3,05
23	2BXF	7-CHLORO-1-METHYL-5-PHENYL-1,3-DIHYDRO-2H- 1,4-BENZODIAZEPIN-2-ONE	2,95
24	2BXG	2-(4-ISOBUTYLPHENYL)PROPIONIC ACID (IBUPROFEN)	2,7
25	2BXI	AZAPROPAZONE; MYRISTIC ACID	2,5
26	2BXX	INDOMETHACIN; AZAPROPAZONE; MYRISTIC ACID	2,4
27	2BXL	2-HYDROXY-3,5-DIODO-BENZOIC ACID; MYRISTIC ACID	2,6
28	2BXM	INDOMETHACIN; MYRISTIC ACID	2,5
29	2BXN	3-[5-[(3-CARBOXY-2,4,6-TRIIODO-PHENYL)CARBAMOYL]PENTANOYLAMINO]- 2,4,6-TRIIODO-BENZOIC ACID; MYRISTIC ACID	2,65
30	2BXO	4-BUTYL-1-(4-HYDROXYPHENYL)-2-PHENYLPYRAZOLIDINE- 3,5-DIONE (OXYPHENBUTAZONE); MYRISTIC ACID	2,6
31	2BXP	4-BUTYL-1,2-DIPHENYL-PYRAZOLIDINE-3,5-DIONE; MYRISTIC ACID	2,3
32	2BXQ	INDOMETHACIN; 4-BUTYL-1,2-DIPHENYL-PYRAZOLIDINE-3,5-DIONE; MYRISTIC ACID	2,6
33	2I2Z	MYRISTIC ACID; 2-HYDROXYBENZOIC ACID (SALICYLIC ACID)	2,7
34	2I30	MYRISTIC ACID; 2-HYDROXYBENZOIC ACID (SALICYLIC ACID)	2,9
35	2XSI	DANSYL-L-GLUTAMATE; MYRISTIC ACID	2,7
36	2XVV	DANSYL-L-ASPARAGINE; MYRISTIC ACID	2,4
37	2XVW	DANSYL-L-ARGININE; MYRISTIC ACID	2,65
38	3B9L	3'-AZIDO-3'-DEOXYTHYMIDINE (AZIDOTHYMIDINE); MYRISTIC ACID	2,6
39	3B9M	3'-AZIDO-3'-DEOXYTHYMIDINE (AZIDOTHYMIDINE); MYRISTIC ACID; 2-HYDROXYBENZOIC ACID (SALICYLIC ACID)	2,7
40	3CX9	(2S)-3-[(R)-(2-AMINOETHOXY)(HYDROXY)PHOSPHORYL]OXY- 2-HYDROXYPROPYL HEXADECANOATE; MYRISTIC ACID	2,8
41	3JRY	SULFATE ION	2,3
42	3LU6	[(1R,2R)-2-[(5-FLUORO-1H-INDOL-2-YL)CARBONYL]AMINO- 2,3-DIHYDRO-1H-INDEN-1-YL]ACETIC ACID	2,7
43	3LU7	4-[(1R,2R)-2-[(5-FLUORO-1H-INDOL-2-YL)CARBONYL]AMINO- 2,3-DIHYDRO-1H-INDEN-1-YL]BUTANOIC ACID; PHOSPHATE ION	2,8
44	3SQJ	MYRISTIC ACID	2,05
45	3UIV	MYRISTIC ACID; (3S,5S,7S)-TRICYCLO[3.3.1.1 3,7]DECAN-1- AMINE (AMANTADINE)	2,2
46	4BKE	PALMITIC ACID	2,35
47	4G03	N/A	2,22
48	4G04	N/A	2,3
49	4IW2	ALPHA-D-GLUCOSE; D-GLUCOSE IN LINEAR FORM; PHOSPHATE ION	2,41
50	4K2C	N/A	3,23
51	4L8U	(2S)-2-[1-AMINO-8-(HYDROXYMETHYL)-9-OXO-9,11- DIHYDROINDOLIZINO[1,2-B]QUINOLIN-7-YL]-2- HYDROXYBUTANOIC ACID; MYRISTIC ACID	2,01
52	4L9K	(2S)-2-HYDROXY-2-[8-(HYDROXYMETHYL)-9-OXO- 9,11-DIHYDROINDOLIZINO[1,2-B]QUINOLIN-7-YL]BUTANOIC ACID	2,4
53	4L9Q	TENIPOSIDE	2,7
54	4LA0	R-BICALUTAMIDE	2,4
55	4LB2	IDARUBICIN	2,8
56	4LB9	ETOPOSIDE; MYRISTIC ACID	2,7
57	4Z69	2-[2,6-DICHLOROPHENYL]AMINO]BENZENEACETIC ACID (DICLOFENAC); PALMITIC ACID; PENTADECANOIC ACID	2,19
58	5IFO	MYRISTIC ACID; RUTHENIUM ION	3,2

Supplementary Table 1. Ligands, resolution and literature references of the HSA dataset.

#	PDB	bound fatty acids	Qh tree	RMSD Tree	RCA cluster
1	1BM0	no	A	A	A
2	1E78	no	A	A	A
3	1E7A	no	A	A	A
4	1E7C	yes	B	B	B
5	1E7E	yes	B	B	B
6	1E7F	yes	B	B	B
7	1E7G	yes	B	B	B
8	1E7H	yes	B	B	B
9	1E7I	yes	B	B	B
10	1GNI	yes	B	B	B
11	1GNJ	yes	B	B	B
12	1H9Z	yes	B	B	B
13	1HA2	yes	B	B	B
14	1HK4	yes	B	B	B
15	1HK5	yes	B	B	B
16	1N5U	yes	B	B	B
17	1O9X	yes	B	B	B
18	1UOR	no	A	A	A
19	2BX8	no	A	A	A
20	2BXB	no	A	A	A
21	2BXC	no	A	A	A
22	2BXD	no	A	A	A
23	2BXF	no	A	A	A
24	2BXG	no	A	A	A
25	2BXI	yes	B	B	B
26	2B XK	yes	B	B	B
27	2BXL	yes	B	B	B
28	2BXM	yes	B	B	B
29	2BXN	yes	B	B	B
30	2BXO	yes	B	B	B
31	2BXP	yes	B	B	B
32	2BXQ	yes	B	B	B
33	2I2Z	yes	B	B	B
34	2I30	yes	B	B	B
35	2XSI	yes	B	B	B
36	2XVV	yes	B	B	B
37	2XVW	yes	B	B	B
38	3B9L	yes	B	B	B
39	3B9M	yes	B	B	B
40	3CX9	yes	B	B	B
41	3JRY	no	A	A	A
42	3LU6	no	A	A	A
43	3LU7	no	A	A	A
44	3SQJ	yes	B	B	B
45	3UIV	yes	B	B	B
46	4BKE	yes	B	B	B
47	4G03	no	A	A	A
48	4G04	no	A	A	A
49	4IW2	no	A	A	A
50	4K2C	no	A	A	A
51	4L8U	yes	B	B	B
52	4L9K	no	A	A	A
53	4L9Q	no	A	A	A
54	4LA0	no	A	A	A
55	4LB2	no	A	A	A
56	4LB9	yes	B	B	B
57	4Z69	yes	B	B	B
58	5IFO	yes	B	B	B

Supplementary Table 2. HSA dataset clusters.

Numerics

The Python functions below perform the principal component analysis (PCA) and random component analysis (RCA).

```
#import pandas as pd #uncomment if needed
import numpy as np
import matplotlib.pyplot as plt
import pylab as pl
#from scipy import stats #uncomment if needed

def GOE(N):
    """
    Returns a random N x N matrix of the GOE ensemble. !
    """
    m = np.random.standard_normal((N, N))
    m = (m + np.transpose(m))/2
    return m

#This function implements the classical PCA
def PCA(data):
    """Returns the dataset projected onto the principal basis system"""
    data -= data.mean(axis=0)
    R_matrix = np.corrcoef(data.T)
    eig_vals, eig_vecs = np.linalg.eig(R_matrix)
    #sort eigenvalues and eigenvectors in decreasing order
    index = np.argsort(eig_vals)[::-1]
    eig_vecs = eig_vecs[:, index]
    #
    #performs a test on the length of the eigenvectors
    for eigvector in eig_vecs:
        np.testing.assert_array_almost_equal(1.0, np.linalg.norm(eigvector))
    #if the test is not passed, Numpy returns an error message!
    return np.dot(eig_vecs.T, data.T).T, eig_vals

#This function implements the random component analysis
def RCA(data, N):
    """Returns the dataset projected onto a random basis system"""
    data -= data.mean(axis=0)
    R_matrix = GOE(N)
    eig_vals, eig_vecs = np.linalg.eig(R_matrix)
    #sort eigenvalues and eigenvectors in decreasing order
    index = np.argsort(eig_vals)[::-1]
    eig_vecs = eig_vecs[:, index]
    #
    #performs a test on the length of the eigenvectors
    for eigvector in eig_vecs:
        np.testing.assert_array_almost_equal(1.0, np.linalg.norm(eigvector))
```

```
#if the test is not passed, Numpy returns an error message!
return np.dot(eig_vecs.T, data.T).T, eig_vals

def RCA1(data, M):
    """Returns the dataset projected onto a random basis system.
       The random matrix should be determined externally.
    """
    data -= data.mean(axis=0)
    R_matrix = M
    eig_vals, eig_vecs = np.linalg.eig(R_matrix)
    #sort eigenvalues and eigenvectors in decreasing order
    index = np.argsort(eig_vals)[::-1]
    eig_vecs = eig_vecs[:, index]
    #
    #performs a test on the length of the eigenvectors
    for eigvector in eig_vecs:
        np.testing.assert_array_almost_equal(1.0, np.linalg.norm(eigvector))
    #if the test is not passed, Numpy returns an error message!
    return np.dot(eig_vecs.T, data.T).T, eig_vals
```